计数

2019年5月6日 19:18

-
- 群计数

1. 给n珠的项链染n色，求允许旋转的本质不同的方案数（P4980）
   1) Burnside引理：对于每个置换σ，定义**C1(σ)为在置换σ下保持不变的方案数。**
   则：本质不同的排列**方案数=各置换中不变方案数C1(σ)的平均数**
   2) Polya定理：对于每个置换σ，定义**C(σ)为σ的不相交循环节的数量**，m为颜色数。
   则：**m^C(σ)=C1(σ)**，代入上式后得到：本质不同**方案数=各置换σ中(颜色数m^循环节数C(σ))的平均数**
   3) 易知旋转操作的循环节长度len=lcm(n,i)/i，则不相交循环节的数量C=n/len=n*i/lcm(n,i)=gcd(n.i)，于是**ans=Σ[i=1:n] n^gcd(i,n) / n**
   4) gcd优化：设gi=gcd(n,i)，易知所有gi都是n的因数，且每个gi会对n的幂次贡献Σ[i=1:n][gcd(n,i)==gi] = Σ[t=1:n/gi][gcd(n,t*gi)==gi] = Σ[t=1:n/gi][gcd(n,t)==1] = phi(n/gi)次，于是ans=Σ[g|n] phi(n/g) * n^g / n

```
int polya(int n,int m) {   //n=元素数，m=色数
     ll rt=0;
     for(ll i=1; i*i<=n; ++i) {
          if(n%i) continue;
          (rt+=euler(i)*qpow(m,n/i))%=P;
          if(i*i!=n) (rt+=euler(n/i)*qpow(m,i))%=P;
     }//特判一下，避免根号n贡献两次
     return rt*qpow(n,P-2)%P; }
```

2. 第一题的基础上，再加上镜面翻转操作（HDU3923、1817）
   1) 偶数点有：n/2种以对珠翻转的方案，循环节(n+2)/2个（选两个珠子与其余n-2个珠子两两配对）；n/2种对珠子中点翻转的方案，循环节n/2个（中点两边的珠子两两配对）（小心n乘法爆int）
   2) 奇数点有：n种翻转方案，各(n+1)/2个循环节（选一个珠与其他珠两两配对）

```
int polya(ll n,int m) {    //n=元素数，m=色数
     ll rt=0;
     for(ll i=1; i*i<=n; ++i) {
          if(n%i) continue;
          (rt+=euler(i)*qpow(m,n/i))%=P;
          if(i*i!=n) (rt+=euler(n/i)*qpow(m,i))%=P;}
     if(n&1) (rt+=n*qpow(m,n+1>>1))%=P;
     else (rt+=(n>>1)*(qpow(m,n+2>>1)+qpow(m,n>>1)))%=P;
     return rt*qpow(n<<1,P-2)%P; }
```

3. 第一题的基础上，m个珠子各有n*n个面可以染色（CF101873B）
   1) 其实就是相当于色数变成c^(n*n%P)，换汤不换药

```
ll n,m,c; cin>>n>>m>>c;
ll t=qpow(c,n*n%P);
for(ll i=1; i<=m; ++i)
     (ans+=qpow(t,__gcd(i,m)))%=P;
(ans*=qpow(m,P-2))%=P;
```

-
- 容斥例题

4. 标号n点树，相邻点必须异色，求恰好k色的染色方案数（CF101933K）

1) 可能是对称筛容斥，令s(n,k)为n点染不超过k色的方案数，则k色中不超过k-1色的方案数=s(n,k-1) * k C k-1，经容斥，答案=Σ[i=1:k] s(n,k-i) * k C k-I
2) 附：阶乘打表法

```
ll fct[MN],fi[MN];
inline void init(int k) {
    fct[1]=fct[0]=fi[0]=1;
    for__(i,2,k) fct[i]=fct[i-1]*i%P;
    fi[k]=qpow(fct[k],P-2);
    rof_(i,k,1) fi[i-1]=fi[i]*i%P;
    for__(i,1,k) c[k][i]=fct[k]*fi[i]%P*fi[k-i]%P;
}
inline void init(int k) {
    c[k][0]=1;
for__(j,1,k) c[k][j]=c[k][j-1]*(k-j+1)%P*qpow(j,P-2)%P;}
inline ll sol(int n,int k) {      //n点，<=k色方案数
    ll rt=k;      //根有k色
    rt*=qpow(k-1,n-1);  //其他n-1点不能与父亲同色，dfs
return rt%P;}
inline int solve(int n,int k) {
    ll rt=sol(n,k);
    for_(d,1,k) (rt+=(d%2? -1: 1)*c[k][k-d]*sol(n,k-d)%P+P)%=P;
return rt;}
```

5. 二进制枚举容斥例：已知买方便面送n种牌概率，求买几包面能凑齐
    1) 买到a牌的概率是p[a]则，需买包数的（几何概率）数学期望是1/p[a]
    2) 买到a或b牌的包数期望是1/(p[a]+p[b])……以此类推
    3) 由容斥原理，买奇数种的期望的-买偶数种的期望是总的期望
    4) 因为牌数在int位数范围内，所以可以如下计算
    5) rof_(i,(1<<n)-1,0){

```
    int c= 0;
    double s=0;
    for_(j,0,n) if(i>>j&1)
        ++c, s+=p[j];
    if(c&1) ans+= 1/s;
    else ans-= 1/s; }
```

6. 求[a,b]中与n的最大公约数超过1的数的数量（不考虑自己与自己的公约数）
    i. b和n的范围都极大，表面是求公约数，实际上考的是质数筛
    ii. 到根号n为止打表，求出所有可能是因子的质数，轮流除n，实现对n的拆分，注意打到根号n以后要判断还有没有剩，n可能有质因子>sqrt(n)！
    iii. 然后容斥算质数的倍数的数量即可
    iv. int sz= fct.size();//因数总数
    v. rof_(i,(1<<sz)-1,0){   //不容斥全0的情况

```
    ll prdct= 1;        //乘积
    int cnt= 0;                //乘积中的因数数量
    for_(j,0,sz) if(i>>j & 1)
        ++cnt, prdct*= fct[j];
    if(cnt&1) ans+= b/prdct- (a-1)/prdct;      //奇加偶减
    else ans-= b/prdct- (a-1)/prdct;}
```

-
- 分块例题

1. Σ[i=1:n]k%i（P2261）（O（2sqrtn））

1) 引：Σ[i=1:n] n/i的向下取整取值不超过2sqrt(n)种（主观理解：i<sqrtn时显然不超过sqrtn种，i>=sqrtn时，n/i的取值映射到n/i，也不会超过sqrtn种）
2) 引：k/i的取值随i上升而递减，k/i向下取整时，函数图像呈阶梯型，连续一“块”的函数值相同，因此可分块计算
3) 注：k超过n时会发生除以零现象，要特判
4) 注：若题目中k=n，则r其实是不需要取min的
5) 原式= Σ[i=1:n]k-k/i*i = n*k - Σ[i=1:n] k/i*i

```
ans=ll(n)*k, n=min(n,k);
for__(l,1,n){
    int r=min(n,k/(k/l));
    ans-=ll(r-l+1)*(l+r)/2*(k/l);
    l=r;   //下一次处理r+1分块
}
```

2. Σ[i=1:n] Σ[j=1:i] i * (i/j)^j（O（nlogn））（牛客练习赛53B）
   1) 换求和次序为Σ[j=1:n] Σ[i=j:n]，内循环每j个连续i共享同一个i/j，只需求出这连续j个i的和sm，用sm乘以(i/j)^j即可计算共享，注意特判i超过n的情况
   2) 外循环中j从1遍历到n的过程可以O（1）转移(i/j)^j（代码中令idj=i/j）

```
for__(i,1,n) pw[i]=1;
for__(j,1,n){
    for(int idj=1,i=j; i<=n; i+=j,++idj){
        int l=i, r=min(n,l+j-1);
        pw[idj]=pw[idj]*idj%P;
        ans=(ans+pw[idj]*sm(l,r))%P;
    }
}
```

◆

◆ 排列组合例题

1. 卢卡斯定理求组合数（模数P为质数时）（P3807）
   1) 相当于将N和m表示为p进制数，对每一位的N'm'分别求组合数，再累乘

```
inline ll qpow(ll a,ll b,int P){ //a^b%P
    ll ans=1;
    for(;b;b>>=1,a=a*a%P)
        if(b&1) ans=ans*a%P;
return ans;}

ll fct[MN],fi[MN];  //阶乘及其逆元

inline void init(int k,int P){   //打表模P的[1,k]阶乘及其逆元
    fct[0]=1;
    for__(i,1,k) fct[i]=fct[i-1]*i%P;
    if(k<P){
        fi[k]=qpow(fct[k],P-2,P);
        rof__(i,k,1) fi[i-1]=fi[i]*i%P;
    }else{ //k阶乘为0，会把所有逆元都变成0，应从P-1开始
        fi[P-1]=qpow(fct[P-1],P-2,P);
        rof__(i,P-1,1) fi[i-1]=fi[i]*i%P;
    }
}

inline int C(int N,int m,int P){ //C_N^m % P
    if(m>N) return 0;
    return fct[N]*fi[m]%P*fi[N-m]%P;
}

ll lucas(int N,int m,int P){     //递归求C_N^m % P
    if(!m) return 1;
```

```
    return C(N%P,m%P,P)*lucas(N/P,m/P,P)%P;
}

int lucas(int N,int m,int P){    //循环求C_N^m % P
    ll rt=1;
    while(N&&m)
        rt=rt*C(N%P,m%P,P)%P,
        N/=P, m/=P;
    return rt;
}
```

2. ExLucas求任意模数P的组合数（和Lucas没半毛钱关系）（P4720）
   1) 对P分解质因子=Σpi^ki，用CRT合并组合数模各pi^ki的结果
   2) 将组合数变形为(n!/pi^x) * (m!/pi^y) / ((n-m)!/pi^z) * (pi^(x-y-z))解决阶乘模pi^ki的逆元不存在的问题，其中xyz是各阶乘中pi因子的次数
   3) 易知1~n之间有n/pi向下取整个pi的倍数，且这些倍数中还可递归的向下继续找pi的倍数；而非倍数者则是对pi取余得1~pi-1，可分块累乘
   4) 即n!/(pi^x) = f(n) = f(n/pi) * ((∏ i [i=1 : pi^ki] [i%pi>0])^(n/(pi^ki))) * (∏ i [i=n/(pi^ki)*(pi^ki) : pi^ki] [i%pi>0])，递归终点f(0) = 1
   5) 则pi的次数x也是类似的O（log^n_pi）递归g(n)=n/p+g(n/p)
   6) 由欧拉定理，m'模p^k的逆元为m'^(phi(p^k)-1)=m'^(p^(k-1)*(p-1)-1)

```
inline ll qpow(ll a,ll b,int P){ //a^b%P，此题中b可能爆int
    ll ans=1;
    for(;b;b>>=1,a=a*a%P)
        if(b&1) ans=ans*a%P;
return ans;}

inline int g(ll n,int p){ //n!中质因子p的次数
    if(n<p) return 0;
    return n/p+g(n/p,p);
}

int f(ll n,int p,int pk){ //n!/(p^x) % pk，其中x=g(n,p)
    if(n==0) return 1;
    ll s=1, s2=1;//<=pk的分块乘积，>pk的块外乘积
    for(ll i=1; i<=pk; ++i)
        if(i%p) s=s*i%pk;
    s=qpow(s,n/pk,pk);
    for(ll i=n/pk*pk; i<=n; ++i)
        if(i%p) s2=i%pk*s2%pk;
    return f(n/p,p,pk)*s%pk*s2%pk;
}

inline ll c(ll N,ll m,int p,int pk){    //C^m_N % (p^k)
    ll rt=f(N,p,pk);
//    (rt*=exinv(f(m,p,pk),pk))%=pk;
//    (rt*=exinv(f(N-m,p,pk),pk))%=pk;
    (rt*=qpow(f(m,p,pk),pk/p*(p-1)-1,pk))%=pk;
    (rt*=qpow(f(N-m,p,pk),pk/p*(p-1)-1,pk))%=pk;
    (rt*=qpow(p,g(N,p)-g(m,p)-g(N-m,p),pk))%=pk;
    return rt;
}

inline ll crt(ll ai,int p,int pk,int P){      //x%(pi^ki)=ai
    return ai*(P/pk)%P*qpow(P/pk,pk/p*(p-1)-1,pk)%P;
}

int exlucas(ll N,ll m,int P){    //C^m_N % P
    ll rt=0, P2=P;
```

```
        int ed=sqrt(P)+1;
        for__(p,2,ed){
            int pk=1;
            while(P2%p==0) pk*=p, P2/=p;
            if(pk>1) (rt+=crt(c(N,m,p,pk),p,pk,P))%=P;
        }
        if(P2>1) (rt+=crt(c(N,m,P2,P2),P2,P2,P))%=P;
        return rt;
    }
```

3. 狼人杀k人桌开法
    1) 题意：n男m女中任找恰k个人开一桌狼人杀，至少4男，至少2女，求开法数
    2) 已知n和m不超过30，利用杨辉三角形递推式打表即可
    3)
```
for_(i,0,35) c[i][0]= 1;
for_(i,1,35) for_(j,1,35) c[i][j]= c[i-1][j-1]+ c[i-1][j];
cin>>n>>m>>k;
for__(a,4,n)
    if(k-a>1) ans+= c[n][a]* c[m][k-a];
    else break;
cout<<ans;
```

4. (a * x + b * y)^k 的第 n 项的系数%10007
    1) 即kCn * a^n * b^(k-n) %10007
    2)
```
for_(i,0,MN){
    c[i][0]= 1;
    for_(j,1,i) c[i][j]= (c[i-1][j-1] + c[i-1][j]) % p;
    c[i][i]= 1;}
while(~scanf("%d%d%d%d",&a,&b,&k,&n)){
    a= qpow(a,n);
    b= qpow(b,k-n);
    printf("%d\n",a*b%p*c[k][n]%p);}
```